In the Specification:

Please amend the specification as follows:

Page 10, paragraph 1:

According ot the invention, for the resampling done by the texture space resampler 132/532 and/or the screen space resampler 140 a choice exists between at least two distinct resampling algorithms. The algorithms may be convention algorithms. The resampler maps (resamples) a collection of input texls/pixels at given locations in an input grid to corresponding collection of output texls/pixels at locations in an output grid. Typically, the locations in the respective grids differ: a transformation takes place. The transformation may be a simple shift, rotation, magnification/minification, but may also be a perspective transformation. The input an output grids may be the same, but may also differ in location and/or resolution. The collection of pixels/texels involved are typically those that correspond to an object or a primitive used to model (part of_ the object. Since the texture and screen are usually 2D, the resampling in principle is also 2D. The same principles can equally be applied to graphics or video processing for rendering of 3D images. For 3D rendering, the basic processing is also typically performed on 2D images. It is well-known how 3D images can be created using 2D techniques, such as projecting two distinct 2D images on the respective eyes of a viewer. The remainder will focus on 2D processing. Fig. 6A shows an example <u>610</u> of 23D resampling in one pass. In the example <u>610</u> of Fig. 6 the input grid 640, indicated by the regularly spaced dots, is the same as the output grid 630. Also the horizontal and vertical spacing is the same. It will be appreciated that the grids need not be the same and that the horizontal and vertical resolution may be different. Fig. 6<u>A</u> shows a resampling in texture space, indicated by the texel coordinates *u*, *v*. The same principle equally applies to resampling in screen space (typically using *x* and *y* coordinates). In the one pass resampling of Fig. 6A, texels that lie in a primitive (or entire object) 660 are resampled from the corresponding texels that lie within the area 640 of a texture map. A transformation occurs in both directions *u*, *v*. In one pass over the texels of the primitives, resampling occurs in both directions. Fig. 6B illustrates the same transformation but now using a two-pass resampling. In each pass resampling occurs in only one direction. In the example of Fig. 6B, the first pass

resamples the input in the horizontal direction *u*, giving an intermediate picture where the input area 640 is mapped to the area 650. The grid of the intermediate image is indicated using 620. In the second pass the area 650 of the intermediate image is resampled only in the remaining direction (i.e. vertical in this example) to the final area 660 in the output image.

Page 12, paragraph 6:

Fig. 9 shows a block diagram of a computer 900, including a central processing unit 910, a memory 920, a display 930, and a computer graphics system 940 according to the invention. The computer may be a convention al computer, such as personal computer, games console or workstation. The computer graphics system may be implemented using a graphics processor. Such a graphics processor may be operated under control of a program causing the graphics processor to execute the method according to the invention. The program may be fixedly embedded (e.g. in ROM), but may also be loaded from a background memory. In the latter case, the program may be distributed in any suitable for, e.g. using a record carrier, such as a CD-ROM, or wired or wireless communication means, such as Internet. The computer graphics system 940 performs one or more of the functions described with respect to Fig. 1. In particular, the computer graphics system 940 may include or be attached to an accumulation buffer 944, as described above. Each resampler may be associated with a respective accumulation buffer. Preferably, the accumulation buffer is integrated in the same IC as the main graphics processor used for executing the graphics processing system. The accumulation buffer is preferably part of a cache memory. The computer graphics system 940 may also include or be connected to a frame buffer 942 containing the entire image to be displayed on the display 930.

Page 13, paragraph 2:

Fig. 10 shows a block diagram of a video processing system 1000, such as a television, including a central processing unit 1010, a memory 1020, a display 1030, and a video subsystem 1040 according to the invention. The video subsystem 1040 may be a implemented using a graphics or video processor. Such a processor may be operated

under control of a program causing the processor to execute the method according to the invention. The program may be fixedly embedded (e.g. in ROM), but may also be loaded from a background memory. In the latter case, the program may be distributed in any suitable for, e.g. using a record carrier, such as a CD-ROM, or wired or wireless communication means, such as Internet. The video subsystem 1040 performs one or more of the functions described with respect to Fig. 1. In a way as described for Fig. 9, the video subsystem 1040 may include or be attached to an accumulation buffer 1044 to a frame buffer 1042 containing the entire image to be displayed on the display 1030. Typically, the video subsystem 1040 also receives a video stream via a video input 1046. The stream may be received using an internal tuner or may be supplied in an y other suitable way, e.g., via a video connection or digital network from a VCR, DVD player, set top box, etc. The video subsystem 1040 may combine graphical information, typically provided under control of the main processor 1010, with the video stream.